

Chapter 10

Industrial Applications

An automated production system is one in which a process is performed by a number of machines without the direct participation of a human worker. The whole automated system is built from smaller automated units suitably coordinated. Such smaller industrial applications of automation are presented in this chapter, most of which have the form of a project for development rather than the scope of a tutorial problem. Some industrial applications met outside a factory environment are also included. If the reader can create proper industrial automation solutions for all of all these projects, it means that he/she is ready to work as professional engineer in the field of industrial automation.

10.1 Cyclic Operation of Traffic Lights

Figure 10.1 shows the time function of traffic light operation at an intersection for two modes of traffic regulation: normal and night operation. The selection of the operation mode is made manually through a rotational switch, but may also be automatic by a sensor signal or a remote signal received from a traffic control center. Write the required program for the operation of the traffic lights.

10.2 Conveyor System for an Assortment of Objects by Pairs

In the conveyor layout of Figure 10.2, the arrangement of objects in pairs is performed. Each pair is received by the conveyor M_3 and should contain one black (B) and one white (W) object, labeled either as BW or WB. The conveyor M_1 brings objects with a random color. The conveyor M_2 is a specific buffer of four positions, each of which can be placed in a central line M_1M_3 by the corresponding activation of relays C_1 – C_4 that drive four pneumatic cylinders (e.g., the activation of relay C_3 brings position 3 in the central line). Each position of the buffer M_2 accepts only two objects. The movable metallic plate binds the forward movement of a single object. After the pairing of a BW or WB is formed, the metallic plate goes down for 5 s by the activation of relay C_5 and the pair is transferred to M_3 . Of course, there is a mechanical coupling of the M_3 motion with each position of M_2 for the transferring of the pairs. When the position of M_2 in central line

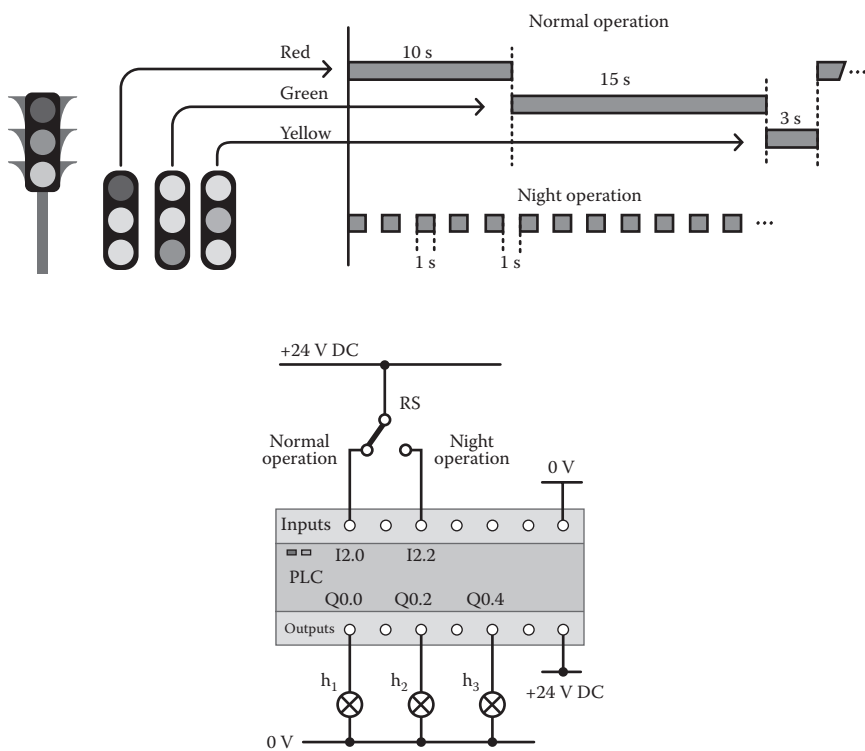


Figure 10.1 Cyclic operation of traffic lights.

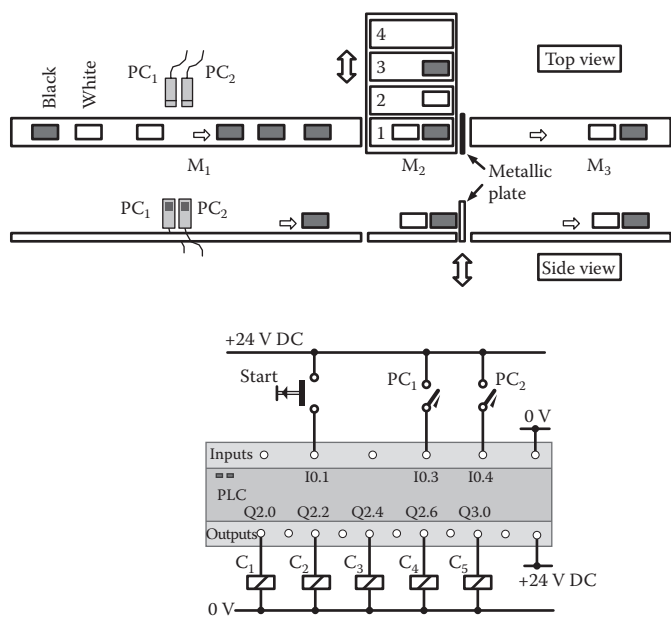


Figure 10.2 Conveyor system for an assortment of objects by pairs.

contains an object, and the next coming object has the same color, then the buffer M_2 must be moved to an empty position or to a position with an object of different color. The arrival of objects is random but suppose that it is impossible to find more than four objects of the same color. The conveyor system starts to operate after a momentary signal from the push of a START button and stops similarly through the push of a STOP button. Write the required program for a PLC.

10.3 Packaging System of Different Balls

Figure 10.3 shows a packaging system of balls in packets of six different items. There are 12 feeding lines, each one with different design. The balls come from all lines in a random way and at relatively slow rate. The packaging system is controlled by a PLC and operates according to the following specifications:

- With the momentary signal START, the machine produces 15 packets of 6 items (90 balls) and waits for the next start signal.
- Each ball, coming from the corresponding feeding pipe, falls on an inclined plane only if the solenoid SL_i is energized, and is detected by the photo-switch F_i . After detection, the solenoid is deactivated in order to block the falling of the same second ball, since the packet must contain different balls ($i = 1-12$).
- Each ball falling on the plane rolls towards to point A and is detected by the photo-switch F_0 . The completion of six balls is verified by the permanent activation of photo-switch F_0 .
- Just after six balls fall, the other balls stop falling, and after the completion of six balls, the wrapping is started at point A. The wrapping is performed by rotating the geared motor M by 360° , and the end of rotation is detected by the proximity switch S_0 .
- If during the production of 90 balls, the number of balls from the same feeding line is greater or equal to 21, then this kind of ball is excluded from the next two sets of 90 balls in production and is reverted normally in the following set of 90.

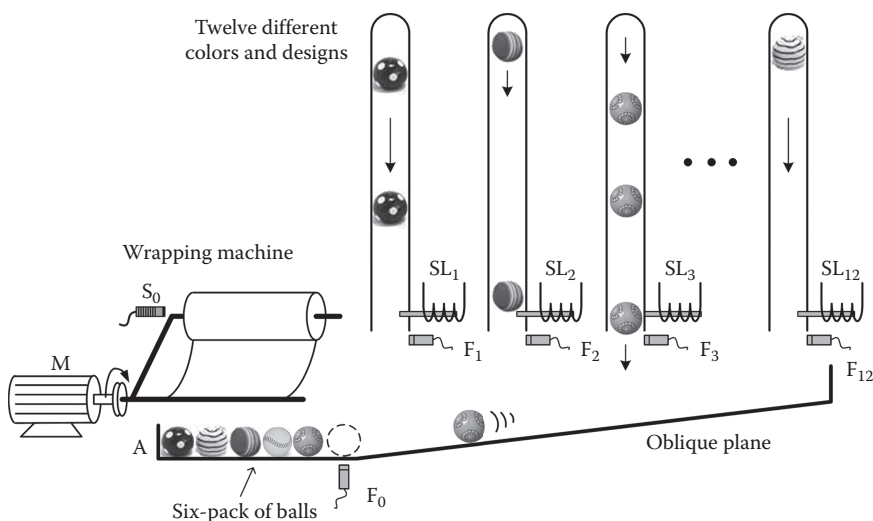


Figure 10.3 Packaging system of different balls.

After making a diagram of the I/O devices connection to a PLC, the required program may be written in Boolean or Instruction List language.

10.4 Conveyor System for Transferring Granular Material with Weight Control

Figure 10.4 shows a conveyor system for transferring granular material from a silo to a process machine (not shown in this figure). The silo has an electrically driven outlet door, which is open when the output Q4.0 is energized. The DC motor moves the conveyor belt with three speeds through the connection of the convenient resistors to the motor coils. A weight

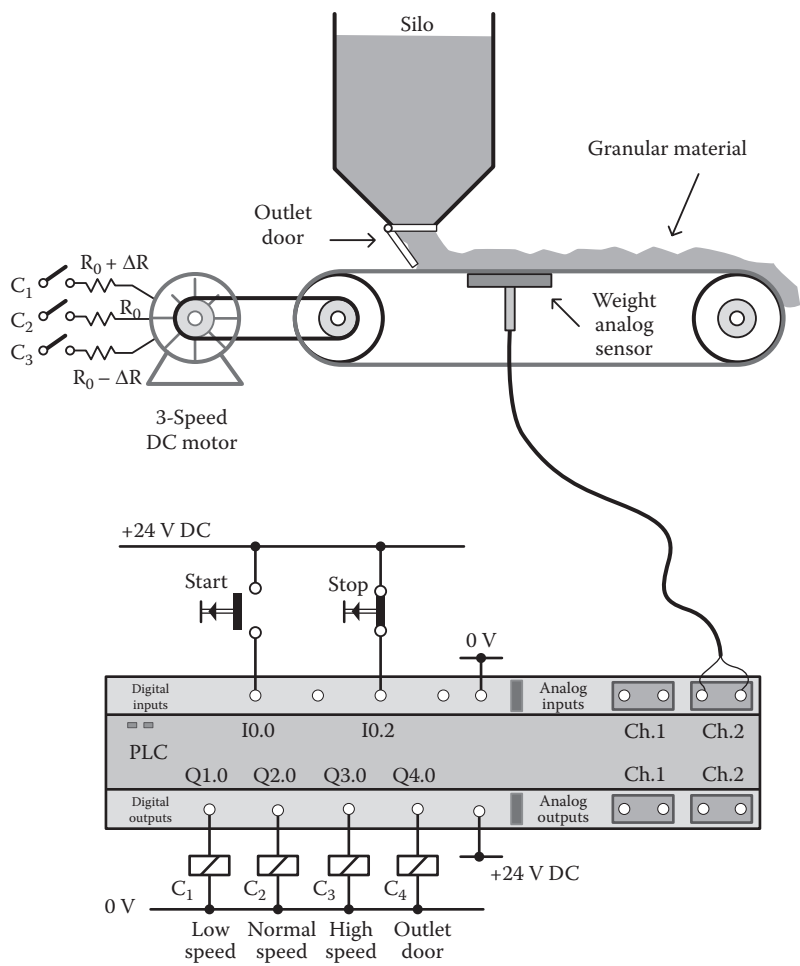


Figure 10.4 Conveyor system for transferring granular material with weight control.

transducer measures the weight of the granular material in the indicated position. It is desired that the conveyor belt operates at normal speed when the measured weight is equal to a pre-defined value of 1 Kg.

The normal speed is achieved by the activation of output Q2.0 (C_2). Due to irregular quantities of the material falling from the silo, the measured weight may vary. When the measured weight is less than 1 Kg, the conveyor speed decreases to “low” by energizing the output Q1.0 (C_1), which adds resistance to the motor coils. If the measured weight is greater than 1 Kg, the conveyor speed increases to “high” by energizing the output Q3.0 (C_3), which subtracts resistance from the motor coils. Obviously, only one of the three outputs connecting the resistors to the motor coils must be energized each time. The conveyor system starts to operate by momentary pressing the START button and then the conveyor belt turns on, and the outlet door opens. By pressing the STOP button at any phase of the operation, the conveyor stops and the door closes immediately. A program for the PLC implementation may regulate the speed of the conveyor belt automatically according to the above specifications.

10.5 The Food Industry: A Machine for Production of Tzatziki Salad

In the food industry, the machine for production of Greek tzatziki (a kind of green salad made from cucumber, yogurt, and garlic) consists of:

- Three silos containing the raw materials
- Three electronic weighing scales with rotating platforms
- A conical tank vessel for mixing materials
- A drum blender
- A dosing pump
- The conveyor belt for packaging of the produced salad, as shown in Figure 10.5a.

Technical characteristics of the equipment.

1. The outlet doors of silos open for feeding the weighing scales via the activation of the pneumatic cylinders C_1 , C_2 , and C_3 , which are extended at the rest state. Each cylinder is retracted by the activation of the corresponding PLC output.
2. Each raw material falls on the corresponding weighing scale with a slow rate of 100 gr/s. The electronic weighing scales have an SPST (NO) output, which is energized at each Kg of weight according to the graph shown in Figure 10.5b.
3. The existent raw material on each scale after weighing, falls in the conical tank vessel via the activation of pneumatic cylinders C_4 , C_5 , and C_6 , which at rest state are also extended.
4. The drum blender and the dosing pump have direct starting motors. At the bottom of the tank vessel there is a level sensor for viscous material with an SPST output that is NO when there is no material in the tank vessel.
5. The conveyor belt operates with a direct starting motor (M_1) and brings the plastic containers to be filled at specific positions.

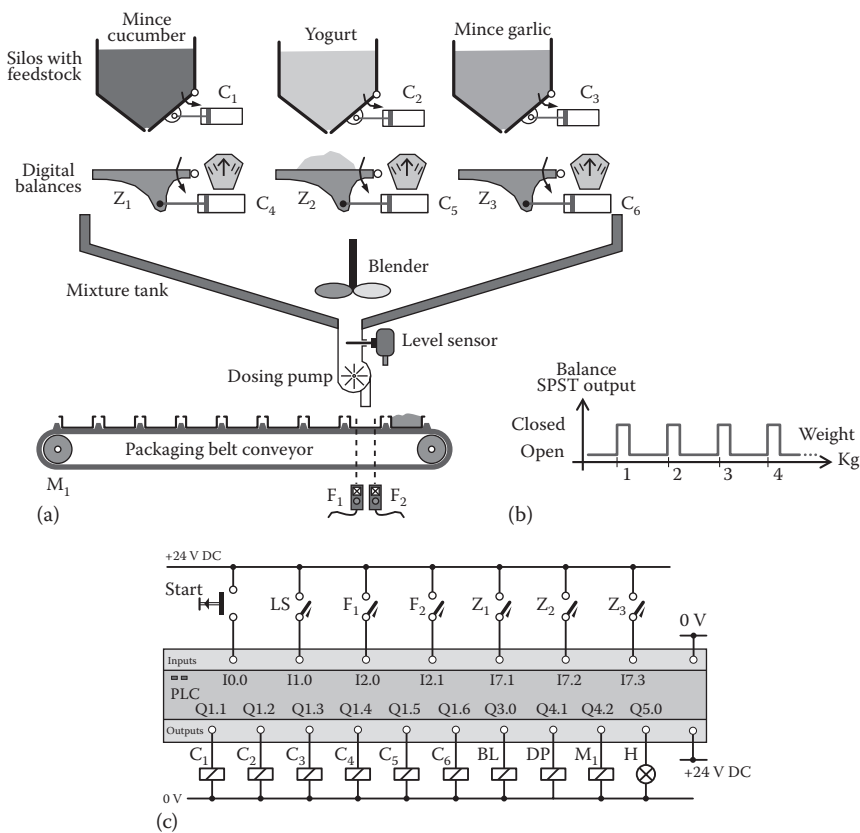


Figure 10.5 Machine for production of tzatziki salad (a), digital balance output (b), I/O device connections in PLC (c).

Operation specifications. The production procedure includes three phases: weighing the raw materials, the feeding and mixing phase, and the packaging phase, which are performed sequentially according to the following specifications:

1. The production process starts with a momentary signal START from the push of a button. Then, the outlet doors of the silos open for feeding weighing scales until the required quantity of material is completed at each scale. After the completion of a material quantity, the corresponding outlet door closes.
2. When the three required quantities have all been completed on the weighing scales, the cylinders C₄, C₅, and C₆ are retracted by activation of the corresponding outputs and cause them to fall in the tank vessel. The three cylinders remain retracted for 5 s in order to be sure that the viscous yogurt has fallen.
3. At the end of the time interval of 5 s, the rotation of the blender starts and keeps on for 12 min.
4. After the end of the mixing procedure, the dosing pump and the conveyor belt start to operate simultaneously. The pump and the conveyor are synchronized by construction during the overall operation.

5. Two photoelectric switches (with SPDT outputs) detect the presence of the plastic container below the outlet of the dosing pump. The two photoelectric switches are necessary due to the small gap between the two successive containers. When there is no plastic container below the pump outlet, the pump stops and operates only the conveyor until it brings the container below the pump outlet. Then the dosing pump starts to operate again.
6. The applied recipe requires 80 Kg yogurt, 20 Kg minced cucumber, and 5 Kg garlic pulp.
7. The filled containers must be counted. The applied recipe corresponds to approximately 300 filled containers. If for any reason there are fewer than 300 filled containers, then the indication lamp H turns on.
8. After the emptying of the tank vessel, all parts of the system stop or reset and the machine is ready for the next START signal.

It is recommended to write the required program in Boolean or IL language and use modular programming.

10.6 Retentive Reciprocating Movement of a Worktable

Retentive operation in a PLC means that in the case of a power interruption, its energized internal elements (M, T, C, and O) preserve their state that existed before the interruption. When the power supply is retrieved, the program of the PLC runs based on the conserved states of these internal elements. In other words, the PLC operation is continuous from the point that it had stopped. Suppose that the memory bits M10.0 to M10.7 have retentive property. The mechanism shown in Figure 10.6 has a worktable that performs a continuous reciprocating movement which (up-down) starts and stops by the same button alternately. The LAD program of the figure only performs the reciprocating logic without the exiting input and does not satisfy a retentive

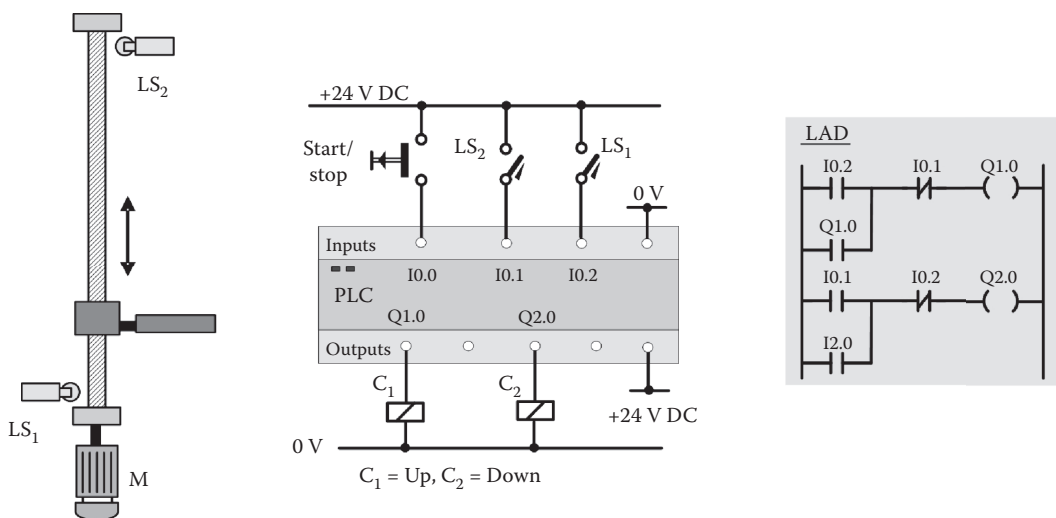


Figure 10.6 Retentive reciprocating movement of a workable.

operation. Write a program for a PLC to achieve the continuous reciprocating movement of the worktable, and present the retentive property, either by modifying the LAD program suitably or writing a new one.

10.7 Wooden Plate Stacking, Painting and Transferring Process

Figure 10.7 shows an arrangement for stacking, painting, and transporting wooden plates. After pressing an ON button, the feeder FD of wooden plates starts to operate. This means that a stack of 12 plates is created at point A, with each plate being counted by the photo switch F_1 . Then the feeder stops and the conveyor belt M operates until it brings the stack from point A to the location B, with the last detected by F_2 . Then the spray painting nozzles (PN_1 and PN_2) are energized for a time interval of 18 s. At the end of the time interval of 18 s, the painting nozzles are deactivated and the conveyor belt transfers the stack to location C, the last detected by F_3 . By pressing the OFF button at any time instant, the operation of the arrangement stops immediately. Write the required program for a PLC.

10.8 An Automated Billiard Table Controlled by a PLC

This application is outside the industrial world, but includes operations very common in industrial processes, such as multi-input counting, signaling, and multiple phases of operation and detection. Figure 10.8 shows an automated billiard table controlled by a small PLC. The specifications of operation are the following:

1. During the game, balls fall from the four corners of the table and are detected by four photo switches F_1 , F_2 , F_3 , and F_4 , while all are gravity-driven to the ball box. Of course, how many balls are going to fall at each corner is a quite random event.
2. The table operates with a coin slot machine. In order for a coin to be accepted by the coin box, it is necessary for button b to be pressed, and all nine balls to be present in the ball box. Also, the solenoid SL must be energized in order for the coin to be driven to the coin box. If for any reason the SL is not energized, the coin comes out in the coin return slot.
3. Just after the detection of a coin by the photo switch F_5 , the SL must be deactivated because during the game, the slot machine does not accept other coins but returns them to the player.

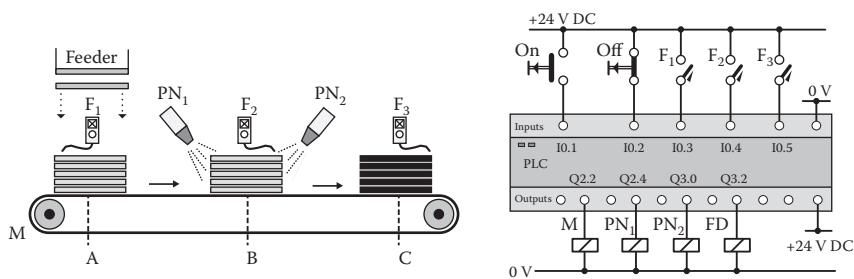


Figure 10.7 Wooden plate stacking, painting, and transferring process.

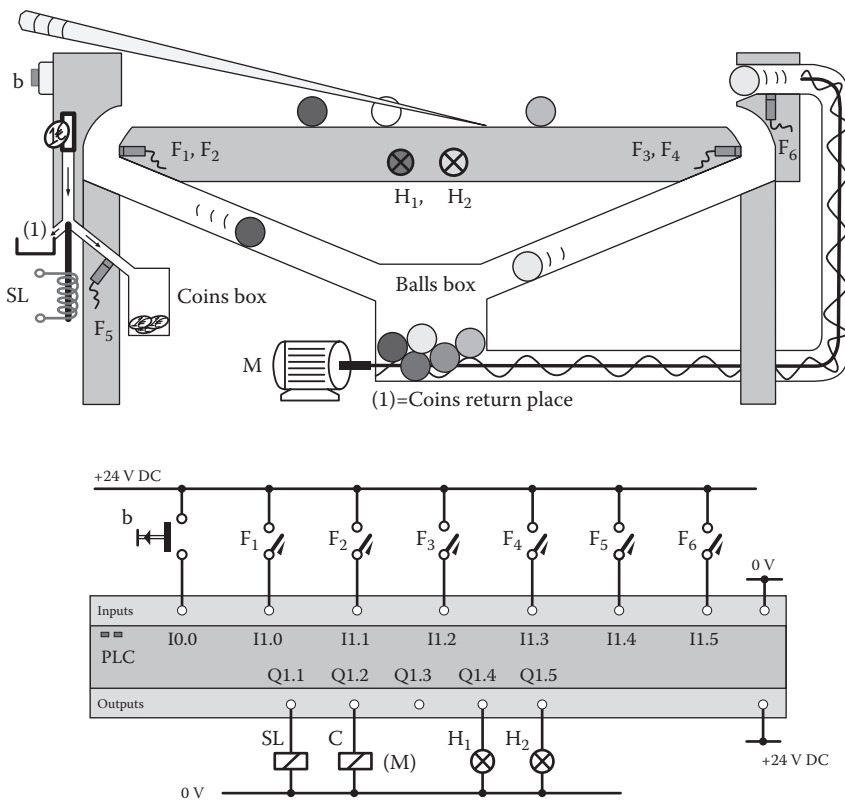


Figure 10.8 Automated billiard table controlled by a PLC.

4. With the same conditions in Step 3, two more actions must take place:
 - a. The motor M starts to operate, the balls are detected and counted by the photo switch F_6 and sent to the table. The motor M stops when the F_6 detects nine balls.
 - b. The red indicating lamp H_1 turns on, which means that the game is in progress.
5. When nine balls have fallen in the ball box, the red lamp turns off and the green lamp H_2 turns on, which means a new game can be started.

The required program has to be written for the I/O addressing shown in the figure.

10.9 Automated Filling of Two Milk Tanks

In a milk factory, a PLC automatically fills two milk tanks. As shown in Figure 10.9, each milk tank has two floaters switches, up and down, which provide the signals “fill” and “empty” respectively. The specifications of operation are the following:

1. The automatic filling of the tanks is energized/de-energized by pressing the ON and OFF buttons respectively. The indication lamp H_0 shows the operation state.
2. Only one milk tank may be filled at a time. If the two tanks are simultaneously empty, then the first tank fills first.

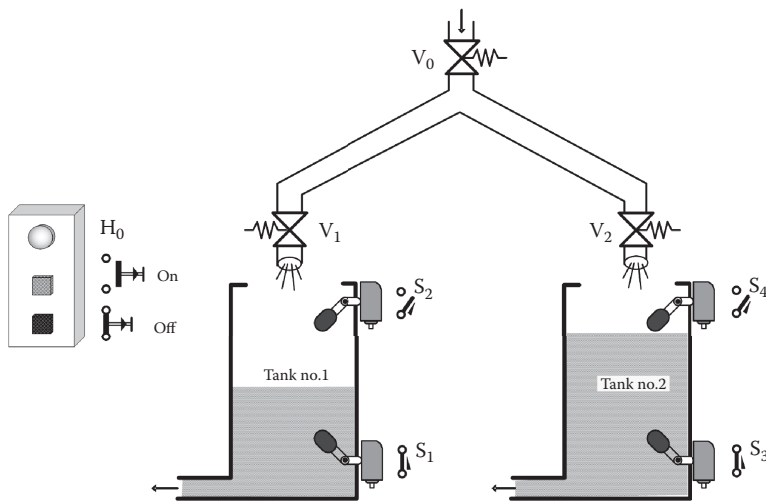


Figure 10.9 Two milk tanks with automatic filling.

3. When the PLC receives an “empty” signal (S_1 or S_3), then the electrovalve V_1 or V_2 opens immediately and the central electrovalve V_0 opens after 5 s. Then, when the signal “fill” (S_2 or S_4) is received, the electrovalve V_0 closes immediately and the electrovalve V_1 or V_2 closes after 10 s.
4. All electrovalves close immediately after the press of the OFF button.

Design a diagram with the connection statuses of the I/O devices and the corresponding addresses, and subsequently, write the required PLC program.

10.10 Modular Programming for a Set of Processing and Repairing Stations

A production line includes six processing stations and one repairing station, each one of which performs its own process or assembly as the produced device passes through them. Each station operates according to its automation program P_i if this is called (executed) by the PLC. The sensor S_2 detects possible defects of the produced device. In such a case, the device is forwarded to the repairing station that operates according to the program P_{123} , while stations 1 to 6 stop operating. When the repairing procedure has finished and the repaired device is ready to reinsert in the central production line (something that is expressed by the activation of sensor S_4), then the operation of stations 1, 2, and 3 must be bypassed. This means that only stations 4, 5, and 6 must start their operation or, in other words, the repaired device does not need processing in stations 1, 2, and 3. Sensor S_1 signals the arrival of a device for processing and assembly. Sensor S_3 signals the end of a device processing. The operation of the conveyor system is not examined here, as also it is considered that a new device does not come before the end of the previous device processing. It is ideal to develop a modular program for the stations' operation, called organization block, including all sub-programs P_i , $i = 1-6$ and P_{123} (Figure 10.10).

10.12 Combined Operation of Two Conveyor Belts

The two conveyor belts M_1 and M_2 , as shown in Figure 10.12, transfer boxes toward a packaging machine. Each has a photocell sensor for counting the transferred boxes. Write a program in Boolean language so that the arithmetic difference of transferred boxes remains less than ten. When this condition is violated, the conveyor belt, having transferred more boxes, must stop until the difference is null, and then the overall process should start again.

10.13 A Machine with Three Operational States and an Acknowledgement Signal

A complex machine has three operational states: stoppage, preheating, and operation. If the machine is in stoppage state, the pressing of a START-STOP button energizes the message “Preheating-Wait” and the output of heating resistors. By pressing the START-STOP button for a second time and if the time interval of three minutes has passed, the motor C is energized, while the message indication and the resistors are de-energized. If during the preheating phase the cancel button is pressed, the message indication and resistors are de-energized. If during this operation the cancel button is pressed, nothing happens, while the START-STOP button is pressed the motor stops immediately. Write a PLC program to satisfy the desired operation specifications (Figure 10.13).

10.14 Chemical Cleaning Process of Metallic Objects

Figure 10.14 shows a tank for chemical cleaning of metallic objects. The basket with metallic objects must be immersed inside the tank several times according to the following plan. By instantaneously pressing the button b_0 , the basket must be immersed three times into the tank and to remain there for 20 seconds each time. Between the three immersions, the basket returns to its initial (up) position and goes down again without delay through the forward/reverse motor M_1 . After the third immersion, the basket returns to the up position and remains there until the next

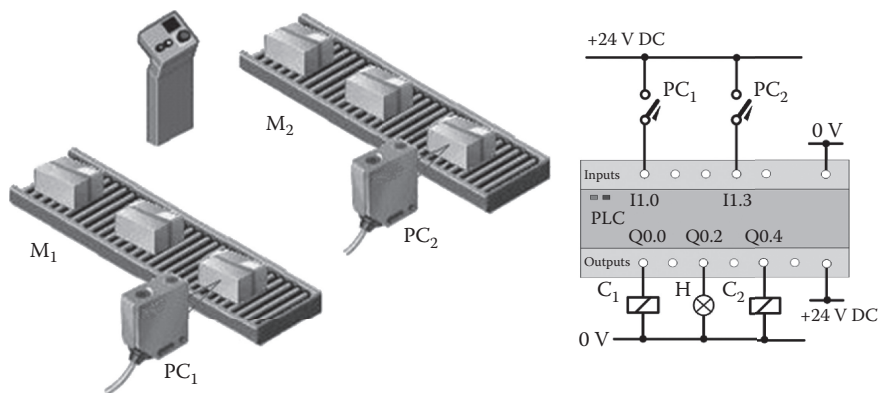


Figure 10.12 Combined operation of two conveyor belts.

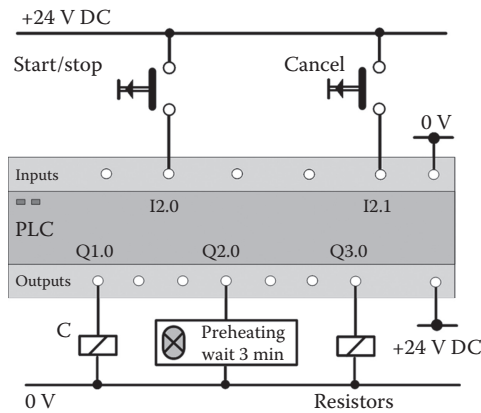


Figure 10.13 I/O device connections to PLC for the application of Section 10.13.

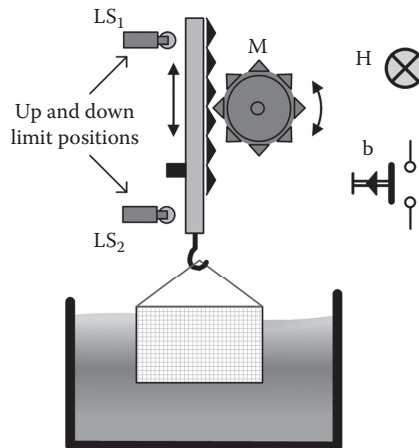


Figure 10.14 Tank-elevator system for chemical cleaning of metallic objects.

pressing of the button b_0 . The indication light H_0 is turned on during the cleaning procedure. After making a diagram of the connection of the I/O devices to a PLC, write the required program in the LAD language.

10.15 Driving a Step Motor Through a PLC

Step motors are electric motors of specific construction for high-precision positioning applications in complex machines. In particular, step motors are brushless DC motors that rotate in discrete steps usually of 1.8° and in both directions, while they can operate in full or half step mode. Step motors require an electronic circuit called a controller or driver to energize the motor phases in a timely sequence to make the motor turn. As shown in Figure 10.15, the stepper driver accepts the following four control signals:

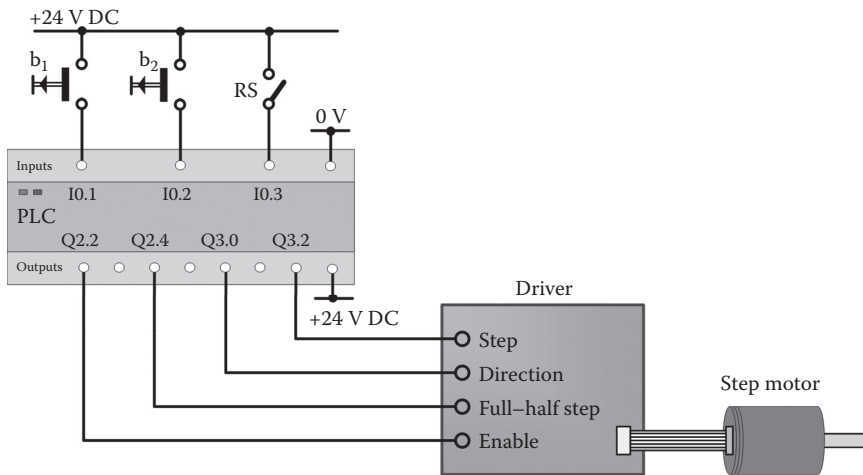


Figure 10.15 The step motor is controlled by the PLC through a driver.

Step. For each pulse applied to the step input of the driver, the motor turns one step ahead. For a series of pulses the motor turns by an equal number of steps. The frequency of the pulse waveform defines the rotation speed of the step motor.

Direction. A logic high signal applied to the direction input of the driver determines one direction, while a logic low signal causes the opposite direction of rotation.

Full half step. The full or half step mode of rotation is determined by a logic high or low signal respectively.

Enable. A logic high signal applied to the enabled input of the driver enables the motor to rotate if the previous three signals have been applied. A logic low signal disables the step motor that cannot rotate, even if the previous three signals are applied.

The above signals can be produced from a PLC equipped with a suitable digital output module. Write a program in Boolean language so that the following operations are achieved:

1. With the press of button b_1 and the RS switch open, the motor rotates left with a full step.
2. With the press of button b_2 and the RS switch open, the motor rotates right with a full step.
3. With the press of button b_1 and the RS switch closed, the motor rotates left with a half step.
4. With the press of button b_2 and the RS switch closed, the motor rotates right with a half step.

The desired speed of rotation corresponds to a pulse frequency of 25 Hz.

10.16 Stacking Machine of Light Objects

Suppose that a step motor is used in a stacking machine of light objects. The controller of the step motor has to save the number of objects that have already been stacked each time in its memory, in order to stack the next object. It is desired to develop such a controller for the motions in a PLC. Using the program written for the problem presented in Section 10.15,

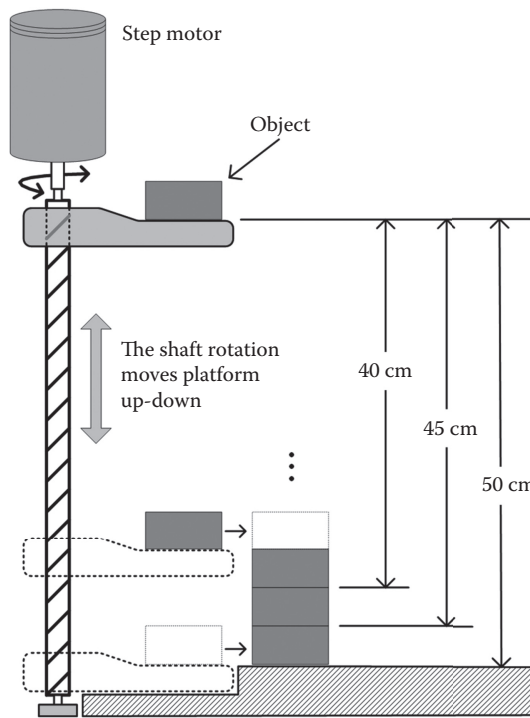


Figure 10.16 Stacking machine of light objects.

write a complementary one so that the PLC accomplishes the sequence of the step motor motions shown in Figure 10.16 and specified as follows:

- With the first pressing of b_0 , the platform goes down 50 cm and returns to its upper location.
- With the second pressing of b_0 , the platform goes down 45 cm and returns to its upper location.
- With the third pressing of b_0 , the platform goes down 40 cm and returns to its upper location.
- With the tenth pressing of b_0 the platform goes down 5 cm and returns to its upper location.
- With the eleventh pressing of b_0 the platform, as in first pressing, goes down 50 cm and returns to its upper location.

Suppose that the rotation of motor axis by 1° causes a linear displacement of the platform of 1 mm. The b_0 is an NO button connected to an additive input I1.0 (not shown) of the PLC, while the outputs are as shown in Figure 10.15.

10.17 A Simple Robotic Arm for Pickup and Placement of Light Objects

A simple robotic station for receiving and stacking light objects is shown in Figure 10.17. For the needs of this industrial automation, suppose that all the movements of the robot are achieved with step motors. The whole system is controlled by a PLC, the I/O connections of which are shown in

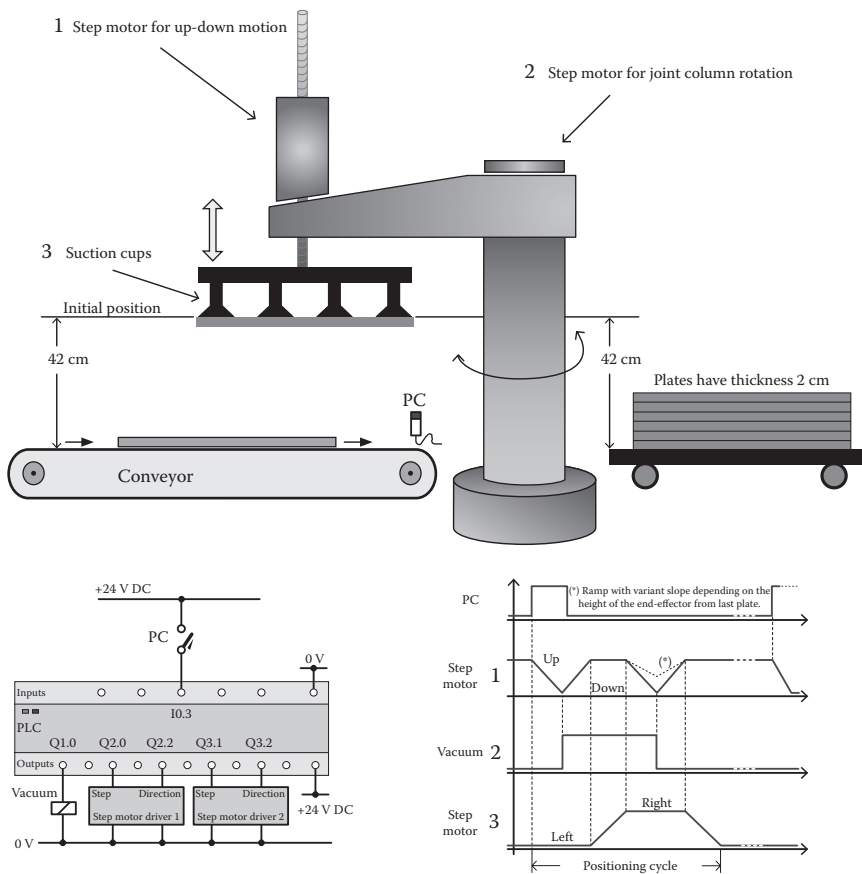


Figure 10.17 Robotic arm for pickup and placement of light objects.

the figure. Using the programs developed in the previous applications in Sections 10.15 and 10.16, write a PLC program so that each arrived object (signal from the photocell PC) is grasped by the robot and placed on the pallet, creating a stack of 20 objects. Particularly at each signal PC, the diagram of the robot movements shown in the figure must be accomplished, as well as the vacuum creation for suction cup feeding.

10.18 Heat Treatment Process in a Chamber Furnace

In heat treatment processes, a gas-fired chamber furnace is usually required. The sliding door of such an industrial furnace, shown in Figure 10.18, is controlled by a PLC. The furnace door, at its open and closed states, activates the proximity switches PS_1 and PS_2 respectively, while the normal state is considered the closed one.

Opening door. The door opens by the momentary pressing of button b_0 through the forward/reverse motor (M) operation. When the sensor PS_1 is activated, the door is open and the motor M stops.

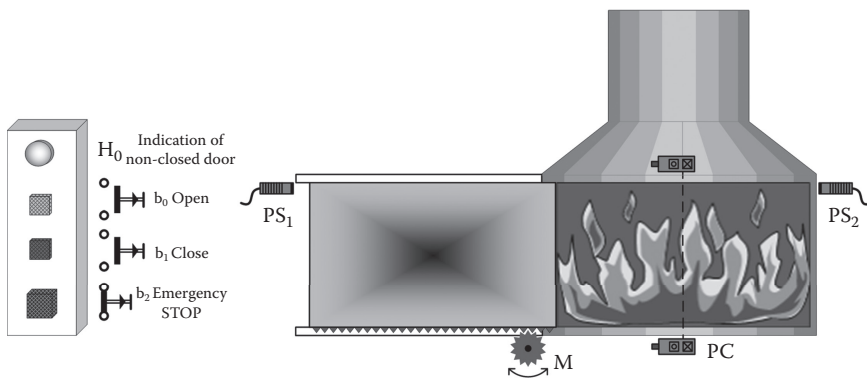


Figure 10.18 Chamber furnace for heat treatment.

Closing door. The door closes automatically after 30 s. The door can also close manually before the completion time of 30 s, by momentary pressing of button b_1 . The indication light H_0 turns on either during the door movement or by being in the open state.

Emergency actions. If during the door closes, the photocell PC is activated, the door stops immediately, while the door continues to close if the photocell is deactivated. The movement of the door, either opening or closing, is interrupted permanently by the pressing of emergency button b_2 .

After making a diagram of the I/O device connections to a PLC, write the required program.

10.19 Working Time Monitoring of a Machine under a Three-Shift Schedule

An industrial plant operates in a three-shift schedule. A complex machine of the production procedure operates intermittently due to the kind of process performed. Its operation also depends on the mood of the machine operator for its working time. Therefore, it is desired to know (e.g., weekly) the total hours of the machine's operation. For this reason, there are three control panels, one for each shift work, and three hour-meters that record the hours of the machine operation per shift work, as shown in Figure 10.19. Each operator can start up the machine only from the control panel corresponding to his shift work if:

- The operator turns on the switch with a key
- The corresponding START button is pressed

Then the machine operates and the hour meter records the time of operation. Any handling from the other two control panels must cause no action. The machine operation stops either by the operator by pressing the STOP button, or by a sensor detecting a critical temperature. After defining the I/O devices and their connections to a PLC, write the required program. Examine also, if it is possible to have the same operation with only one pair of START-STOP buttons common for the three shift works.

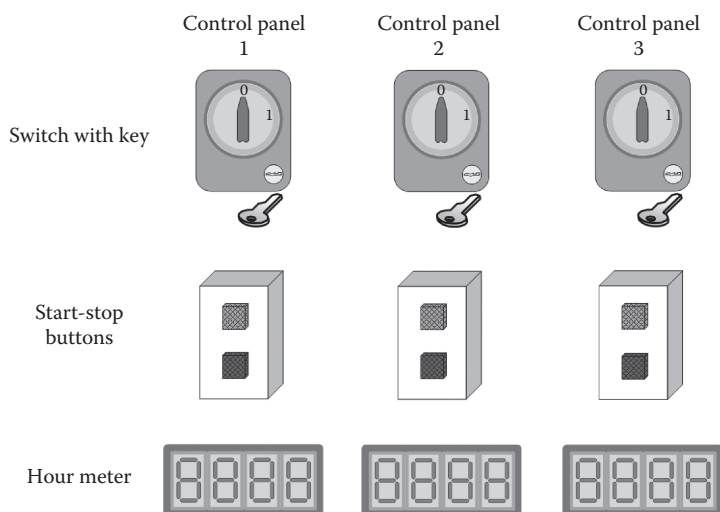


Figure 10.19 Working time monitoring of a machine.

10.20 Feeding an Assembly Machine with Components in Bulk

Figure 10.20 shows an arrangement for supplying an assembly machine with metallic components of Π schema that exist in bulk in a bucket. The rotating disk D has four curved blades, and each one can accept a single component which, due to blade rotation, is thrown on the pickup rail and simultaneously detected by the proximity switch PS. The component collection by any of the blades is a completely random event that may or may not happen one or more times. The assembly machine might need 4, 8, or 12 parts depending on the assembled device, which is selected by the operator by pressing the corresponding push-buttons b_1 , b_2 , and b_3 .

The specifications of operation are the following:

1. The system is set ready for operation with the switch RS₀₋₁, otherwise nothing can work.
2. Subsequently, the operator must choose the desired number of parts, which is marked on the control panel.
3. With the condition that the number of parts has been selected, and by pressing momentary the START button, the disk rotation begins and continues to feed the selected number of components, and then stops.

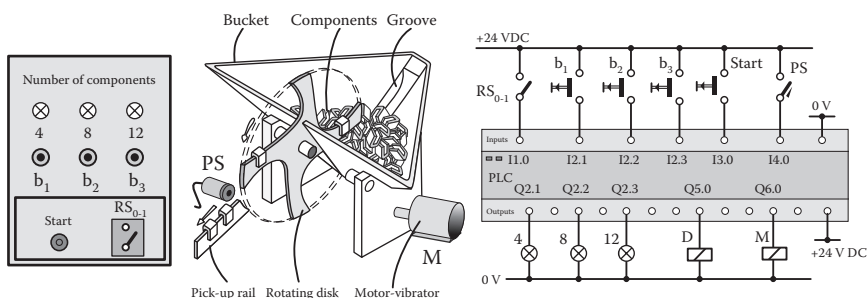


Figure 10.20 Feeder system of an assembly machine.

4. With a new START command, Step 3 is repeated, and so on.
5. If during the disk rotation it is spent without feeding a component for 2 min, then the vibrator motor M is automatically switched on for 10 s (without stopping the rotation of the disc), which shakes the bucket and the components, and thus facilitates their collection by a blade.
6. If during a feeding procedure (after START) a different number of components is selected, the system does not respond to this change. Only in the non-feeding phase (the disk does not rotate) can the desired number of components be changed.

If at any time the switch $RS_{0.1}$ is opened, the system stops and all are reset.

Write the required PLC program in Boolean language.

10.21 A Roller Conveyor System for Wrapping Plastic Membrane

Figure 10.21 shows part of a roller conveyor for transferring a plastic membrane from a production machine to a wrapping machine. In this section of the conveyor, a floating folding is created deliberately, in order to have the potential difference between the rate of production and the rate of wrapping absorbed. The length of folding is measured with a laser-type photocell (PC) for distance measurement and should be kept between 40 cm and 50 cm limits through the regulation of the speed of the motor M_1 , which performs the wrapping. The analog output 0–10 V of the distance meter is stored in the memory location MW200 of a PLC. With the START and STOP buttons, both motors M_1 and M_2 of the roller conveyor start and stop, respectively. The operating speed of the M_1 is regulated so that when:

Folding < 40 cm $\Rightarrow$ Low speed

Folding > 50 cm $\Rightarrow$ High speed

40 cm $\leq$ Folding $\leq$ 50 cm $\Rightarrow$ Normal speed

In addition to the main programming task, tasks also include an additive digital output (not shown in the PLC) to be activated when the desired limits are violated, and write the overall PLC program.

10.22 Color-Based Separation of Plastic Balls

Figure 10.22 shows a mechanism that separates plastic balls depending on their color and packs them inside boxes in quantities of 6 or 12 pieces, depending on the position of the selector switch SS. The color sensor S_C detects the color of a ball (yellow or green) by measuring the light rays emitted by the motionless ball. The recognition time of a color from the S_C is 1 s. The activation of coil 1 happens after the S_C signal and allows the ball to drop. After the last pass from the proximity switch S_p , coil 1 is deactivated in order to stop the next ball in front of the sensor S_C . Coil 2 sets where each ball will fall according to its color, and the movable bearing has the position shown in the figure when coil 2 is deactivated.

When the chosen quantity is completed in every box, the corresponding indication is activated, and remains on until the next ball of the same color falls. The mechanism comes into operation (or pause) via the switch $RS_{0.1}$. The removal of the filled boxes is automatically handled by another machine and is not part of this application. Write the required PLC program in Boolean language.

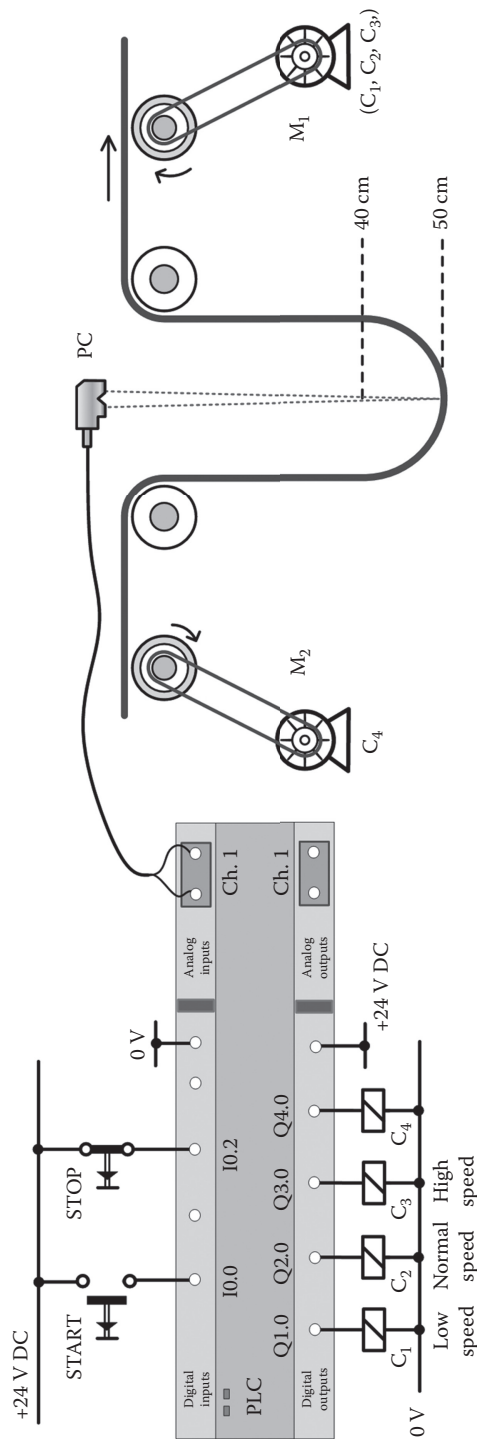


Figure 10.21 Roller conveyor for wrapping plastic membrane.

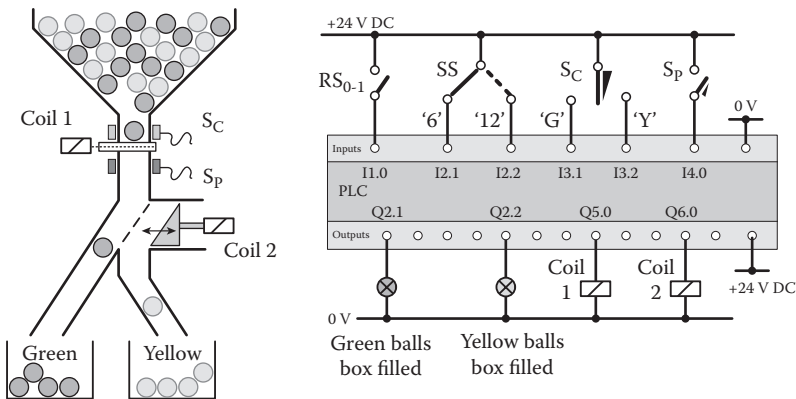


Figure 10.22 Color-based separation of plastic balls.

10.23 The Shearing Machine of an Unfolded Aluminum Sheet

A machine M unfolds and launches aluminum sheet in a laminate cutting table in order to be cut by a shearing machine SHM into pieces of a specified length. The machine M can drive the metal sheet at two speeds; at a low speed by the activation of the relay C_1 , and at a high speed by the activation of C_2 . The length of the metal sheet is measured via a pulse encoder ENC which is rotating during the sheet motion. The correspondence between pulses and sheet length is 1 pulse = 1 mm. It is desired for the machine to cut two possible lengths: 50 cm and 100 cm (Figure 10.23).

Operating specifications:

1. The length selection is done via the selector switch SS_{1-2} .
2. The pieces of small (large) length are cut at a low (high) speed, respectively.
3. When the desired length is achieved, the machine M stops and the shearing machine operates for 3 s. After 3 s passes, the cutting procedure continues by starting again the machine M.
4. With the START signal the operation of the whole machine begins, while with the STOP signal it stops. If during the operation of the machine at any cutting length, someone changes the position of the selector switch SS_{1-2} , there should be no change.

Write the required PLC program in any desired language.

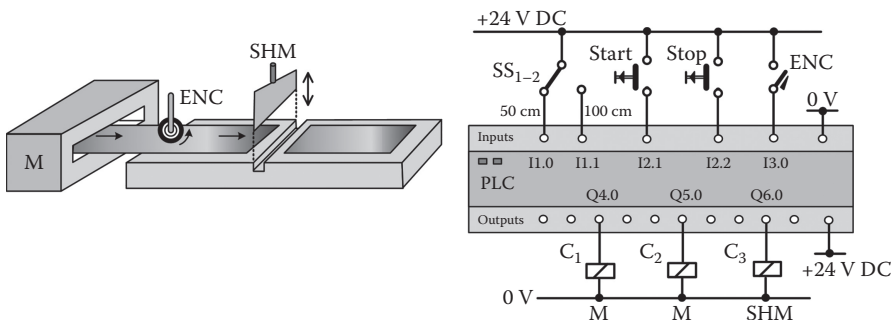


Figure 10.23 Shearing machine of an unfolded aluminum sheet.

10.24 Inlet and Outlet Rate Monitoring of Transferred Objects In and Out of a Process Machine

The transportation system of a processing station, which includes two independent conveyor belts M_1 and M_2 , is shown in Figure 10.24. The first conveyor part inserts objects in the station and the second part removes the processed objects. Two sensors PS_1 and PS_2 detect the incoming and outgoing items respectively. Write an automation program so that the PLC stores the current counting of processed objects, based on what will regulate the operation of the transportation system and specifically:

1. The conveyor belt M_1 starts to operate via the switch RS_{0-1} .
2. If there are objects, one or more, in the processing station, then M_2 also operates.
3. If there are more than five objects inside the station, then the M_1 stops, and starts automatically again when objects become less than five.
4. If there are no objects in the station, then M_2 stops.
5. If the conveyor M_1 has stopped for a period exceeding 30 s, then the current counting is zeroed, considering that in the existing station the objects have been destroyed.
6. With the switch RS_{0-1} open, both conveyors stop.

10.25 A Metal Plate Rolling Mill Machine and Control of Their Thickness

Figure 10.25 shows an arrangement for rolling metal plates. The hot metal is pushed and squeezed by two driving rollers A and B through which a metal sheet with thickness $d = 5$ mm (under ideal conditions) is produced. The compression roller A may apply (simultaneously with the rotation) three different forces F_{-1A} , F_{0A} , and F_{+1A} via the activation of relays C_1 , C_2 , and C_3 respectively, and for which $F_{-1A} < F_{0A} < F_{+1A}$ is valid. The compression roller B may apply two different forces, F_{0B} and F_{+1B} , via the activation of the relays C_4 and C_5 , respectively, and for which $F_{0B} < F_{+1B}$ is valid. The thickness of the metal sheet is measured with a suitable analog sensor, the measurement of

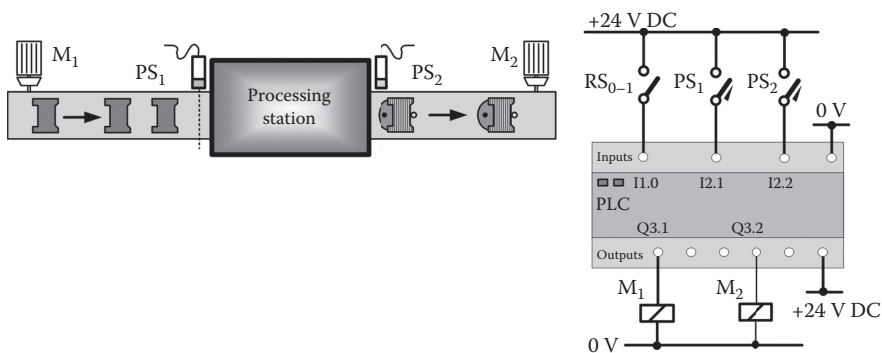


Figure 10.24 Rate monitoring of transferred objects in a process machine.

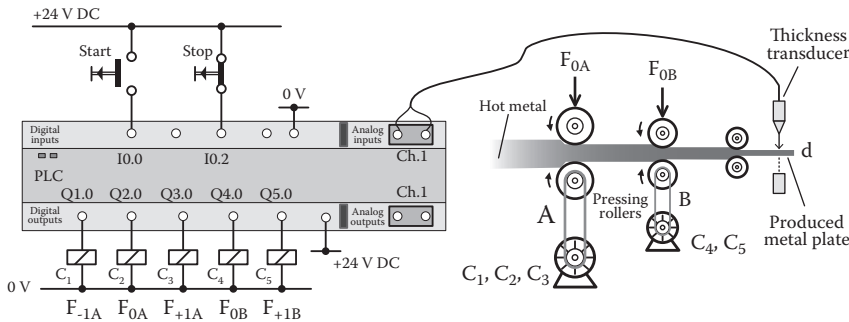


Figure 10.25 Metal plate rolling mill machine.

which is stored in the word memory MW100 of the PLC. Write the required PLC program in Boolean language that will adjust the thickness of the produced sheet (the conditions are never optimal) in accordance with the following specifications:

1. When the measured thickness is $d = 5$ mm, then the nominal forces F_{0A} and F_{0B} are applied.
2. When $5 < d \leq 5.1$ mm, the applied force is increased in roller B.
3. When $d > 5.1$ mm, the applied force is increased in both rollers.
4. When $d < 5$ mm, the applied force is decreased in roller A, while roller B applies the nominal force.
5. The operation starts by pressing the START button, and initially the nominal forces F_{0A} and F_{0B} are applied. By pressing the STOP button, both rollers cease to operate.

10.26 An Object Painting and Transporting System

Figure 10.26 shows a painting layout of objects that are transported on the conveyor belt M_1 . For the color-dye injection system to function properly, the air pressure inside the tank should be between the limit values of 3 and 4 bar. The air pressure inside the tank is measured with the help of the pressure transducer P (the measured value is stored in MW 200—analogue input 0–10 V) and is regulated by a simple ON-OFF control of compressor M_2 . This means that:

- If $P \geq 4$ bar, then $M_2 = \text{OFF}$.
 - If $P < 3$ bar, then $M_2 = \text{ON}$ and remains ON until $P = 4$ bar.
1. The whole layout is put into operation (or pause) via the switch $RS_{0.1}$.
 2. The conveyor belt M_1 operates continuously (if $RS_{0.1} = \text{ON}$) until an object is moved under the painting nozzle and is detected by the photocell PC, then it stops.
 3. On the condition that the pressure is 3–4 bar and the compressor M_2 is not operating, the valve V is energized for 3 s, which results in the spraying of color liquid on the object, i.e., painting the object.
 4. After the elapse of 3 s, the valve V closes and the conveyor belt M_1 starts to operate again, until the next object arrives and the same process is repeated.

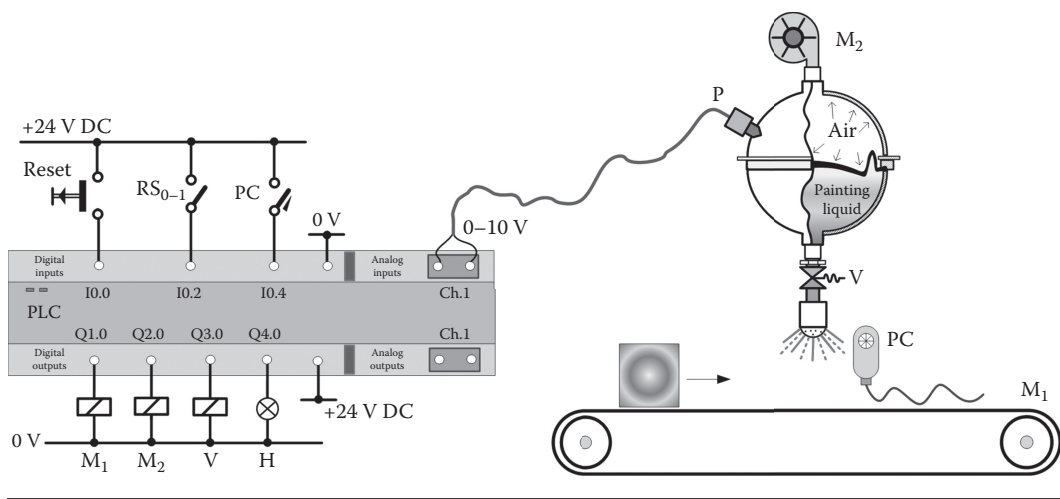


Figure 10.26 Object pointing and transporting system.

5. When 100 objects have been painted, the indicator H lights turn on and the conveyor belt stops. The pressure inside the tank continues to be controlled. By pressing the reset button, the conveyor belt starts again for the painting of the second 100 objects and so on.

The painting layout is controlled by a PLC, and for this operation, the overall automation should be programmed in a preferable language.

10.27 A Multiple Bottle Packing Station

A bottle packing station can accept bottles from three different bottling lines (M_1 – M_3) that produce different products, as shown in Figure 10.27. The station packs bottles each time from one bottling line only, and they are forwarded toward it through the central conveyor belt M_0 . Slat chain conveyors M_1 – M_3 of bottling lines operate continuously and transfer bottles regardless of whether these are packaged or not. The movement of bottles that are not packaged is blocked from the corresponding movable barriers, whose open position is detected via the limit switches b_1 – b_3 and is caused by the activation of the solenoids So_1 – So_3 . In order to avoid an error in the packaging procedure with products of different type on the same crate, the conveyor M_0 operates automatically (without a START-STOP command) if only one barrier is open. If for any reason, two or three barriers are found opened at the same time, the conveyor M_0 should immediately be stopped and the relative indicator H (meaning wrong packaging) should be activated. In such a case, the continuation of the packaging process is achieved by the intervention of an operator, which will remove the wrong bottles and manually will close the barrier that caused the problem.

When a crate has been fed with 12 bottles, then the conveyor belt M_4 must be enabled for 4 s in order for the filled crate to be removed, and a new empty crate to be placed from another machine, the operation of which does not interest us. During the time interval of 4 s, the conveyor belt M_0 remains stopped.

The packaging process starts with the closing of the switch RS_{0-1} . The opening of the latter results in an interruption of operation only if the current packaging order has been completed.

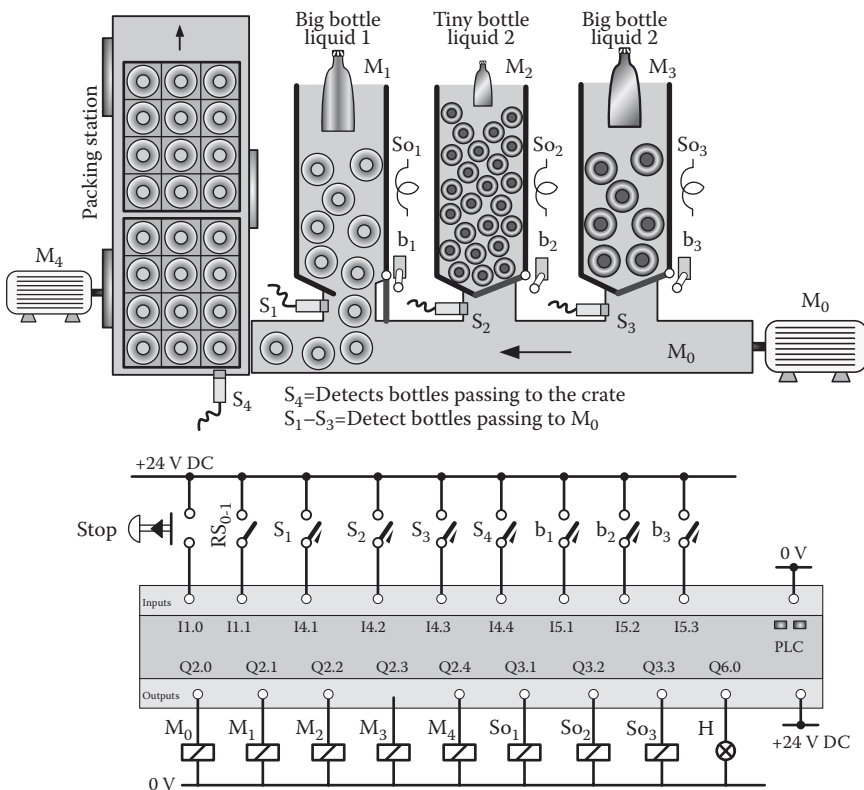


Figure 10.27 Multiple bottle packing station.

If any operating accident happens, then the pressing of the pushbutton emergency stop causes a general discontinuation, regardless of the point at which the packaging process is running.

The packaging process refers to the complete execution of a packaging order including a triple of crates from all the produced products (e.g., n_1 , n_2 , and n_3), depending on each customer's desire. This means that after closing the switch RS_{0-1} , the station will have to pack n_1 crates with products from M_1 , n_2 crates with products from M_2 , and n_3 crates with products from M_3 .

A PLC program for this industrial application should be parametric to n_1 , n_2 , and n_3 . As a first approach to the problem however, you are prompted to write the required program that implements a specific order such as (2, 1, and 3).

10.28 A Barrel-Filling System for Dry Bulk Material

Figure 10.28 shows a dry bulk material barrel-filling system. The filling process involves the transportation of barrels through the conveyor belt, the automatic positioning of the barrels, and their filling based on a level detection. The sequence of actions to fill a barrel is as follows:

1. Initially, the lights in the standby indicator (after putting PLC in run mode) turn on.
2. The filling process is activated by pressing the START button. Then the conveyor belt starts to operate (motor) by bringing empty barrels and the run indicator turns on, while

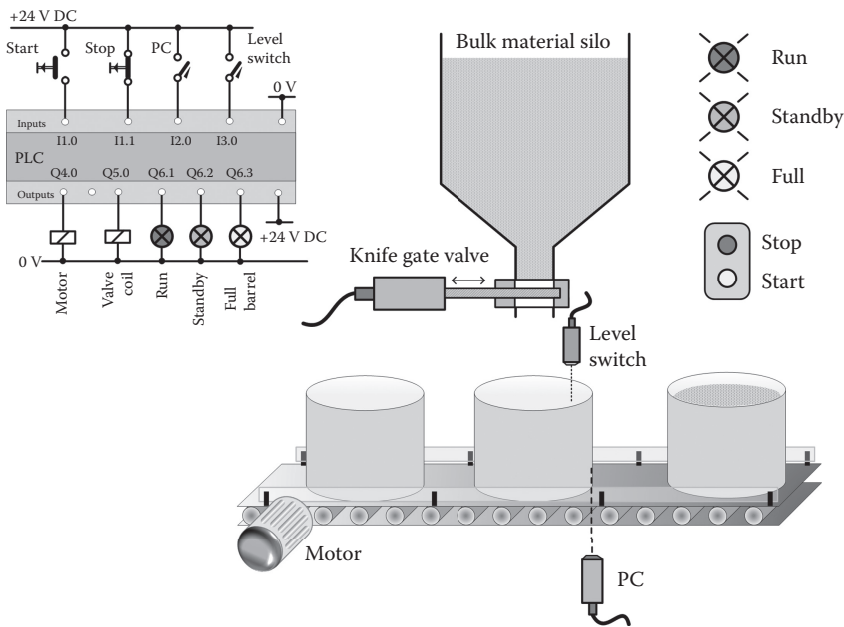


Figure 10.28 Barrel-filling machine for dry bulk material.

the standby indicator turns off. The filling process and the conveyor belt cease to operate if the STOP button is pressed. Then the run and standby indicators change their status of indication.

3. When a barrel reaches the proper position, which is detected by the photocell PC, the conveyor belt stops. With the barrel in the correct position, and the conveyor belt stopped, the knife gate valve is energized, and the dry bulk material falls into the barrel.
4. The knife gate valve will be disabled and the filling of the barrel will stop when the level switch is energized. Then, the indicator full turns on and will remain lit until the filled barrel is removed from the PC.
5. Once a barrel is filled (level sensor signal), the conveyor belt operates again in order to remove the filled barrel and to bring the next empty barrel, and so on. The next barrel arrival is a timely random event.

This automation problem can be further extended by adding the need for the kind of product and filling recipe selection in the PLC program.

10.29 An Electro-Pneumatic System for Pickup and Lay Down of Plastic Containers

Figure 10.29 shows a pickup and lay down mechanism of plastic cheese storage containers. The mechanism includes a special swivel bracket M_1 that initially is in a horizontal position, a pneumatic valve V for creating suction (vacuum), two proximity switches that detect the horizontal

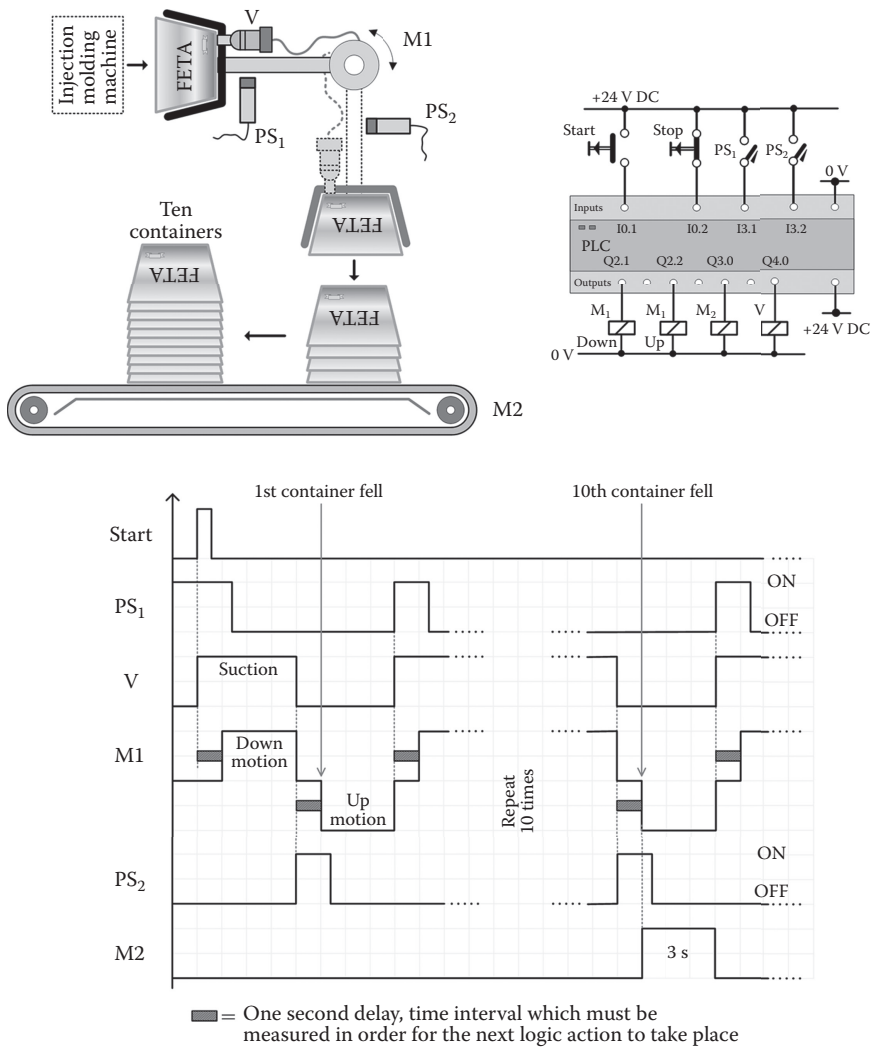


Figure 10.29 Electro-pneumatic system for pickup and lay down of plastic containers.

and vertical position of the bracket M_1 , and the conveyor belt M_2 for deposition and removal of container stacks. The sequence of evolved operations is as follows:

- With the bracket in horizontal position and by pressing the START button, a vacuum is created ($V=On$).
- The plastic container is sucked from the injection molding machine. Following a delay of 1 s, the bracket (M_1) moves downward, and once detected by PS_2 , the vacuum stops ($V=Off$).
- Following a delay of 1 s, the bracket moves upwards.
- Once detected by the PS_1 the vacuum is activated again ($V=On$), and the same process is repeated 10 times.

- The conveyor belt M_2 operates for a time window of 3 s, removing the stack of 10 plastic containers that has been created.
- After 3 s passes, the entire process is repeated to create the second stack of containers and so on.
- After pressing the STOP button, at any current stage of the operation, the mechanism stops and the bracket comes to a horizontal position.

The exact sequence of the individual actions is illustrated in detail in the diagram shown in Figure 10.29. It is recommended that the required PLC program be written in Boolean or IL language.

Appendix A: Arithmetic Systems

A.1 Introduction to Arithmetic Systems

Although knowledge related to arithmetic systems is the subject of books with mathematical or computer content, some basic issues of arithmetic systems are included in this book because it is absolutely necessary for an understanding of the operation and programming of PLCs. On the other hand, the reader's knowledge of arithmetic systems is usually satisfactory from basic computer courses, and it is much more necessary in using PLCs than personal computers. Even the most numerate user of a PC does not often come into contact with the concept of a "bit" since there is no such need. In contrast, in the world of PLCs, the programmer of an automation system starts with the utilization of bits, continued by integers, real numbers, or hexadecimal numbers, and ends again with the utilization of bits. The basic reason for this is the fact that the majority of devices that a PLC controls are of a binary status (binary digits or bits, such as 0 or 1, ON or OFF). Moreover, there are indication devices operating in a BCD code, where they are used, among others, in the octal system for the input/output address and corresponding program instructions for bit to bit operations, etc. Therefore, someone who knows the arithmetic systems and codes better will have an easier time programming and making use of even the most specialized instructions.

An arithmetic system is a code of symbols which are assigned a quantity. From the moment the code has been defined and memorized, it can be used to measure any quantity. The decimal arithmetic system uses ten basic (primordial) symbols or digits, where each digit represents a certain quantity. When several digits are grouped together, then larger quantities can be measured. The base of an arithmetic system is called the number of the primordial symbols or digits of the system. All arithmetic systems use position weighting to express the significance of each digit in a group.

A.2 Decimal Arithmetic System

In a decimal arithmetic system, ten unique digits or numbers are used, from 0 to 9, and hence the base is 10. The value of a decimal number depends on the amount of digits and the position weight of each digit. The first position to the immediate left of the decimal point corresponds to units, which follows that of decades, hundreds, etc., as shown in Figure A.1. Each position value can be expressed as a power of 10, starting from 10^0 to the left and 10^{-1} to the right of the decimal point.

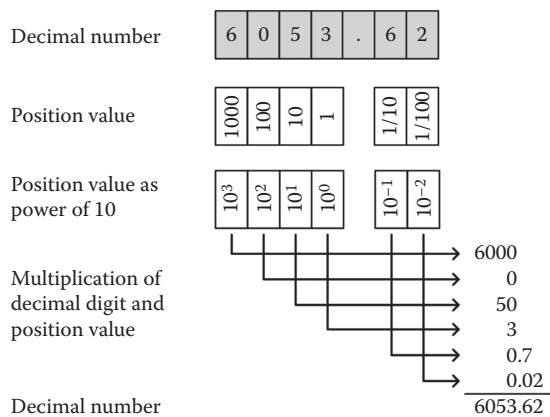


Figure A.1 Decimal number system.

A.3 Binary Arithmetic System

The binary arithmetic system uses the two digits 0 and 1, so its base is 2. Any number can be expressed as a combination of digits 0 and 1. In order for an electronic circuit to process numbers, which is called a “digital circuit”, it is necessary to join the numbers with corresponding electric voltage signals. Therefore, if a digital device, such as the PLC, were to process decimal numbers, then ten different voltages would be required, one voltage level for each digit. Consequently, each decimal number will be expressed by a combination of these ten voltage levels. The implementation of a digital circuit for processing decimal numbers, due to many electric elements needed for creating the ten different voltages, will lead to a very complex circuitry. For this reason, to avoid complex electronic circuits, the binary arithmetic system is the most suitable for developing digital circuits for processing numbers, since only two voltage levels are required. Assuming that 5 V and 0 V represent the digits 1 and 0 correspondingly, a four-digit binary number (such as 1011) will be represented electronically by the combination of voltages 5 V, 0 V, 5 V, and 5 V, which can be recognized by a processor. It is widely known that a combination of eight binary digits (bits) is called a “byte” and a combination of 16 binary digits is called a “word”.

The position weighting that is assigned to each digit of the binary number, due to the base 2, will be doubled for each step to the left of the number, with the first right position to correspond to the value 1 (2^0). As in the case of the decimal arithmetic system, all the position values of the binary arithmetic system can be expressed as powers of two, as shown in Figure A.2. The digit of

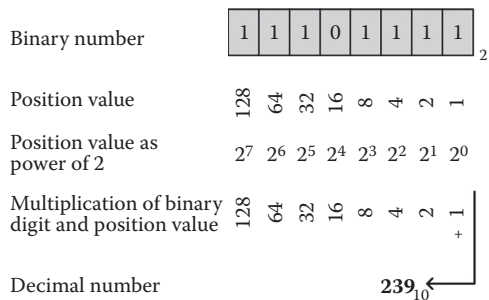


Figure A.2 Binary system of numbers and their conversion to the decimal system.

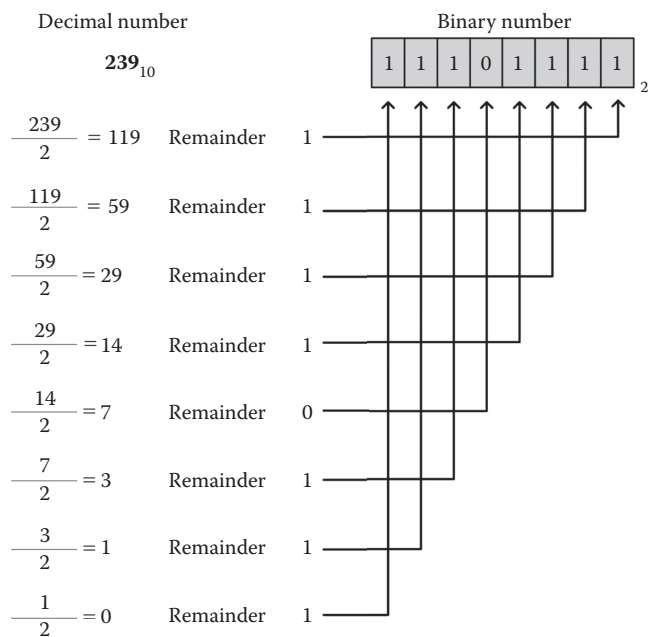


Figure A.3 Conversion of a decimal number to binary.

a binary number with the lowest position value is called the “least significant bit” or LSB, while the digit with the highest position value is called the “most significant bit” or MSB. In order to convert a decimal number to a binary one, the procedure illustrated in Figure A.3 is followed. This means dividing successively the decimal number by 2 and denoting the remainders 0 or 1 as digits of the binary number.

A.3.1 Negative Numbers

The CPU of various PLCs process the analog values in a binary form only. Analog input modules convert the analog process signal into digital form, and analog output modules convert the digital output value into an analog signal. The representation of a digitized analog value depends on the number of used bits (resolution), which has to include both positive and negative numbers. In the decimal arithmetic system, negative or positive numbers are denoted with a minus or plus sign in front of them, which means that each number has two parts: the sign that is a symbol and the magnitude. Therefore, this method of digitized value representation is not applicable in a digital device like the PLC.

The simplest method to represent a negative number as a binary number is to use an extra digit, called a “sign bit”, which usually is the fifteenth bit (MSB) for a 16-bit resolution. If the sign bit is 0 (1), this means that the number is positive (negative), such as the signed binary numbers shown in Figure A.4.

A second method for the representation of negative numbers in a digital device is to use the complement of a binary number, and particularly the one’s complement or two’s complement. When a binary number, has the value 1 the complement is 0 and vice versa, that is called the one’s complement. For example, the one’s complement of the binary number 0110 is 1001.

Binary number		Decimal number
Sign	Magnitude	
0	1 1 1	+7
0	1 1 0	+6
0	1 0 1	+5
0	1 0 0	+4
0	0 1 1	+3
0	0 1 0	+2
0	0 0 1	+1
0	0 0 0	+0
1	0 0 1	-1
1	0 1 0	-2
1	0 1 1	-3
1	1 0 0	-4
1	1 0 1	-5
1	1 1 0	-6
1	1 1 1	-7

Figure A.4 Signed binary numbers.

The magnitude of a negative number represented with regard to one's complement is the one's complement of the corresponding positive number magnitude. Some examples of negative numbers represented in one's complement, are shown in Figure A.5.

The two's complement of a binary number is the sum of one's complement plus 1, as expressed by Equation A.1,

$$2\text{'s complement} = 1\text{'s complement} + 1 \tag{A.1}$$

Binary number		Decimal number	1's complement
Sign	Magnitude		Magnitude
1	0 0 1	-1	1 1 1 0
1	0 1 0	-2	1 1 0 1
1	0 1 1	-3	1 1 0 0
1	1 0 0	-4	1 0 1 1
1	1 0 1	-5	1 0 1 0
1	1 1 0	-6	1 0 0 1
1	1 1 1	-7	1 0 0 0
			↑ Sign

Figure A.5 Representation of negative numbers using the complement of a binary number (one's complement).

PLCs use the two's complement to perform subtraction of two binary numbers, which means that the subtraction is replaced by an additive operation, as clarified by Equation A.2,

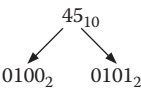
$$(7)_{10} - (5)_{10} = (7)_{10} + (-5)_{10} = \underbrace{(0111)_2 + (1011)_2}_{2\text{'s complement}} = (0010)_2 = (2)_{10}$$

(A.2)

The subtraction operation using the two's complement facilitates its implementation in a PLC, since the same digital circuit will perform both addition and subtraction.

A.4 Binary-Coded Decimal Arithmetic System (BCD)

In a binary-coded decimal arithmetic system, each digit of a decimal number is coded via a four-digit binary number. This definition means that each digit of a decimal number is simply substituted by the corresponding binary number, as in the following example,



In the past, computer systems had specific units for the execution of the arithmetic operations between BCD numbers in order to overcome memory storage problems. In some PLCs today, the BCD arithmetic system is even supported since it is compatible with the operation of some output devices as an indication display device, or input devices as an arithmetic switch. Figure A.6 shows an example of converting a BCD number to an equivalent decimal number.

A.5 Octal Arithmetic System

The octal arithmetic system is used by some manufacturers of PLCs, such as Allen-Bradley and Siemens, for addressing their digital inputs and outputs. The octal system uses the numbers 0 through 7 and hence has a base 8. The position weighting values of the octal system are derived in a similar way as that applied to the binary and decimal systems. Starting from the first right position with value the null power of 8 (8⁰), the value at each next x position increases by the x

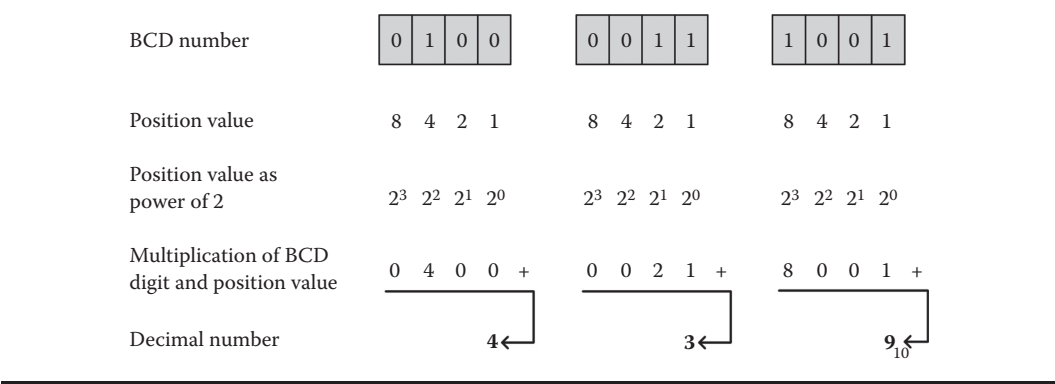


Figure A.6 Conversion of a BCD number to a decimal.

Octal number	<table><tr><td>2</td><td>5</td><td>7</td></tr></table> ₈	2	5	7			
2	5	7					
Position value	64 8 1						
Position value as power of 8	8 ² 8 ¹ 8 ⁰						
Multiplication of octal digit and position value	<table><tr><td>128</td><td>40</td><td>7</td></tr><tr><td></td><td></td><td>+</td></tr></table>	128	40	7			+
128	40	7					
		+					
Decimal number	175 ₁₀						

Figure A.7 Conversion of an octal number to a decimal.

power of 8. Figure A.7 shows an example of converting an octal number to an equivalent decimal number and the three first-position values.

The conversion of a decimal number to an octal number is performed by successive divisions of the decimal number by 8 and denoting the remainders as digits of the octal number. The first remainder corresponds to the least-significant digit of the octal number. An example of such a conversion procedure is the following:

215/8	= 26 with remainder 7
26/8	= 3 with remainder 2
3/8	= 0 with remainder 3, therefore the octal number is 327 ₈ .

Since the largest one-digit number of the octal system is 7, three binary bits are enough to represent each of the eight digits of the octal system. The conversion of an octal number to an equivalent binary number is performed by replacing each octal digit by its 3-bit binary equivalent, as shown in Figure A.8.

The octal arithmetic system is the most user-friendly, since it is harmonized to the 8-bit general digital structure, independently of the 8-bit or 16-bit or 64-bit structure of memories and microprocessors. The manufacturers of large industrial PLCs use the octal arithmetic system to address words and bit locations in their memory as also digital I/O modules usually in groups of 4, 8, and 16 inputs/outputs. Therefore, is easy for a user to know an individual single bit of the memory to which digital input or output corresponds among the various I/O modules.

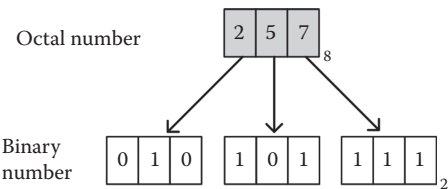


Figure A.8 Conversion of an octal number to binary.

A.6 Hexadecimal Arithmetic System

Another arithmetic system that is widely used in PLCs and PCs is the hexadecimal arithmetic system, also often referred to as HEX. This system has a base 16, which means that it uses 16-digit symbols that take decimal values from 0 to 15. But because it will create confusion if a hexadecimal digit consisted of two numbers, the digits from 10 to 15 are represented by the letters of the alphabet A to F, as shown in Table A.1. For example, the hypothetical number 15_{16} is not clear what expresses, because may be the hexadecimal 15 (equivalent decimal value 15) or two discrete digits, the hexadecimal 1 and the hexadecimal 5, which have an equivalent decimal value of 21. Although the use of letters in the hexadecimal system seems complicated at first sight, the hexadecimal code is the most appropriate for the digital representation of numbers. The reason is that it is possible with two hexadecimal digits to represent all the decimal numbers from 0 to 255 (0 to FF), while in a binary arithmetic system 8 binary bits or 1 byte are required.

The position weighting values of the hexadecimal system are derived in a similar way to that applied to binary and decimal systems. Starting from the first right position with the null power value of 16 (16^0), the value at each position of a hexadecimal digit rises to the left of the LSB by a power of 16. Converting a decimal number to a hexadecimal one becomes with subsequent

Table A.1 The Hexadecimal Digits and Their Equivalents in Both Binary and Decimal Systems

<i>Hexadecimal Number</i>	<i>Binary Number</i>				<i>Decimal Number</i>
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
A	1	0	1	0	10
B	1	0	1	1	11
C	1	1	0	0	12
D	1	1	0	1	13
E	1	1	1	0	14
F	1	1	1	1	15

Hexadecimal number	<table><tr><td>1</td><td>2</td><td>3</td><td>E</td></tr></table> ₁₆	1	2	3	E
1	2	3	E		
Position value	<table><tr><td>4096</td><td>256</td><td>16</td><td>1</td></tr></table>	4096	256	16	1
4096	256	16	1		
Position value as power of 16	<table><tr><td>16³</td><td>16²</td><td>16¹</td><td>16⁰</td></tr></table>	16 ³	16 ²	16 ¹	16 ⁰
16 ³	16 ²	16 ¹	16 ⁰		
Multiplication of hexadecimal digit and position value	<table><tr><td>4096</td><td>512</td><td>48</td><td>+ 14</td></tr></table>	4096	512	48	+ 14
4096	512	48	+ 14		
Decimal number	<table><tr><td colspan="4">4670₁₀</td></tr></table>	4670 ₁₀			
4670 ₁₀					

Figure A.9 Conversion of a hexadecimal number to a decimal.

divisions of the decimal number by 16, and by denoting the remainders of divisions as the digits of the hexadecimal number, as in the following example for decimal 4670:

4670/16	= 291 with remainder 14 (=E)
291/16	= 18 with remainder 3
18/16	= 1 with remainder 2
1/16	= 0 with remainder 1, therefore hexadecimal number is 123E ₁₆

The conversion of a hexadecimal number to a decimal number is performed by a procedure similar to this, which was applied in previous arithmetic systems, such as the one shown in Figure A.9. Finally, the conversion of a hexadecimal number to a binary number is done with a simple replacement of each hexadecimal digit with its equivalent binary form of 4-bits, according to Table A.1.

A.6.1 Parity Checking

Arithmetic digital data, independently of the utilized arithmetic system and the way that they are represented, do not stay static at some memory location of a digital device, but instead they are transported from one point to another. Digital data are transferred continuously, especially in PLCs, mainly due to the real-time control that PLCs perform (such as from a PLC to a peripheral device, from a PLC to another PLC via a communication network, etc.). It is very likely for a single bit of binary data to change value from 1 to 0, due to electromagnetic noise or a transitional phenomenon, or any other imponderable reason. For the proper detection of a possible error during the transmission or storage of binary data, the parity checking technique is applied which consists of adding an extra bit, called a parity bit, in the transferred data word.

Even Parity. The parity bit is added to the transferred data in such way that the total number of units (1) are even. For example, if the byte 01010100 is transmitted, then the parity bit 1 has to be added and becomes 010101001.

Odd Parity. The parity bit is added to the transferred data in such a way that the total number of units (1) are odd. For example, if the byte 01010100 is transmitted, then the parity bit 0 has to be added and becomes 010101000.

When receiving or reading the transmitted data word, the total number of units (1) are checked if they are even or odd, and thus it is determined whether there was an error in data transmission. When establishing a communication connection between two PLCs, the user is usually invited to select an even or odd parity in the corresponding software environment.

A.7 ASCII Code

A digital device, such as a computer or a PLC, should have the ability to process anything, except for numbers and letters, since this functionality is needed for enabling the programming of the PLC with text commands (e.g., an instruction AND). As in the case of numbers being represented based on an arithmetic system in a binary form that can be processed by a PLC, letters should also be represented in binary format. For this purpose, the coding of the letters of the alphabet and other symbols used to write text has been provided by the introduction of the ASCII code (American Standard Code for Information Language). Based on the ASCII code, 7 bits were initially used to represent the various letters and symbols. Since the number of letters and symbols surpassed the maximum 128 (2^7), which could cover the 7 bits, an eighth bit was added. A table with ASCII codes is not listed here, because it is a well-known issue and they appear in many books for someone who would like more details.

A.8 Gray Code

Gray code is a binary encoding method that does not use the position weighting of a digit like the other arithmetic systems do. Gray code defines only the transition from one number to the next where only one bit changes its status, and for this reason is not applicable in any types of arithmetical operations. However, it does have some applications in analog to digital converters and in some input/output devices as the encoders. In binary coding, two or more contiguous bits change their status to express a decimal number incremented by one; for instance, when going from 7 to 8 (0111 to 1000) there are four bits changing their state. In Gray coding, only one bit changes its status to express the same increment. For this reason, Gray code is ideal for use in PLCs and computers. In principle, it is the code that shows the minimum possible error because when only one bit changes from one state to another, the probability of error is drastically reduced. For the same reason, the transmission speed of Gray code is comparatively higher than others such as the BCD code. In Table A.2, Gray 4-bit codes are shown in relation to the equivalent binary codes for comparison purpose.

In the industrial world, automation technology, robotics, and especially in PLCs, Gray code is encountered often because the position encoders that base their operation on it may be input devices of a digital controller or PLC simultaneously. The position encoders (rotary and linear, absolute and incremental) have all been presented in Section 2.3.7. In general, position encoders connected to a PLC apply a pulse in an input module, which follows Gray code, i.e., only one bit changes at each step of a shaft rotation (rotary encoders) or linear movement of a machine carriage (linear encoders).

This appendix will end with a brief reference to the digital representation form of the numbers in PLCs. Generally in computers, the numbers are represented either as fixed-point or

Table A.2 Gray Code and Binary Equivalents

<i>Gray Code</i>				<i>Binary Number</i>			
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

floating-point numbers. In PLCs, both arithmetic possibilities are offered to a user. Specifically, most medium or large PLCs support the processing of:

- Single-precision integers (16-bit numbers with a range of values from -32768 to 32767)
- Double-precision integers (32-bit numbers with a range of values from -2147438648 to 2147438647)
- Floating point real numbers of single precision (32-bit numbers with a range of values from $-3,402823E+38$ to $3,402824E+38$)

It should be noted that the alteration of an arithmetic operation from another one (e.g., for single-precision integers from the corresponding double-precision integers) is performed using different programming instructions, such as the instructions **+I** and **+D** for this example, respectively. In Chapter 7, instructions have not been included for all kinds of numeric representation for the same reasons that have been explained regarding the advisability of the instructions included in Table 2.1.

Appendix B: Analog I/O Values Scaling

B.1 PLC Analog I/O Values Scaling

During the programming task of an automation application, including the processing of analog values, it is necessary to address some secondary issues, such as the violation of limits of an analog input, the scaling-conversion of analog I/O range into units of physical magnitude (also called engineering units), the overflow or underflow of digital arithmetic limits (>32768 or <-32768), etc. The correct or incorrect operation of the PLC program depends on proper knowledge of these issues. In the operation manuals with the technical characteristics of the PLC's analog I/O modules, one can find the required information concerning the digital representation of analog values for all the standard analog ranges that an analog module can accept, the measurement or overload limits of the A/D or D/A converter, and also the manner of address. Here, reference will be provided only to the analog value scaling that is illustrated in Figure B.1, and expresses the need for a human operator to use engineering units.

Let's assume that a temperature transducer has an analog output 0–10 V DC that is connected to an analog input of a PLC. The output voltage of the transducer is proportional to the temperature range 100 °C to 500 °C. The transducer measures the temperature of a thermal process, which should fluctuate from 250 °C to 300 °C by applying the required control. Figure B.2 illustrates the whole scaling procedure, particularly the linear relation between the input (voltage) and the output (arithmetic value that expresses °C). In the diagram, the maximum voltage 10 V corresponds to the arithmetic value 32767, but in some PLCs, this may be 27648 or another arithmetic limit, depending on the existence or absence of an overflow or over-range detection. The input/output linear relation is expressed by the following equations:

$$\text{Scaled Output Value} = (\text{Input Voltage} \times \text{Slope}) + \text{Offset} \quad (\text{B.1})$$

$$\text{Slope} = \frac{\text{Max Scaled Value} - \text{Min Scaled Value}}{\text{Max Input Value} - \text{Min Scaled Value}} \quad (\text{B.2})$$

$$\text{Offset} = \text{Min Scaled Value} - (\text{Min Input} \times \text{Slope}) \quad (\text{B.3})$$

What is needed in the sequence, is to determine which voltage values correspond to the temperature limits (250 °C and 300 °C) that are controlled in order not to be violated. The calculation is based on Equation B.1, which is solved for “Input Voltage”,

$$\text{Input Voltage} = (\text{Scaled Output Value} - \text{Offset}) / \text{Slope} \quad (\text{B.4})$$

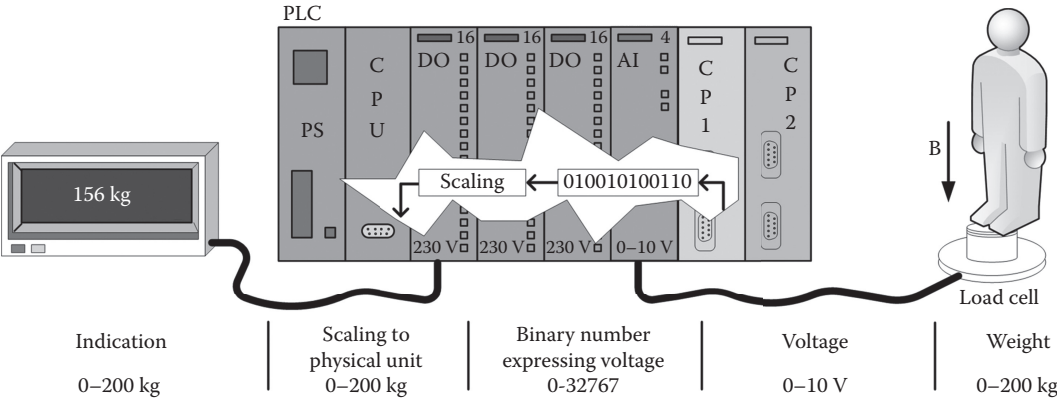


Figure B.1 The scaling of an analog value is a necessary procedure to convert numeric data into meaningful units of the corresponding physical variable.

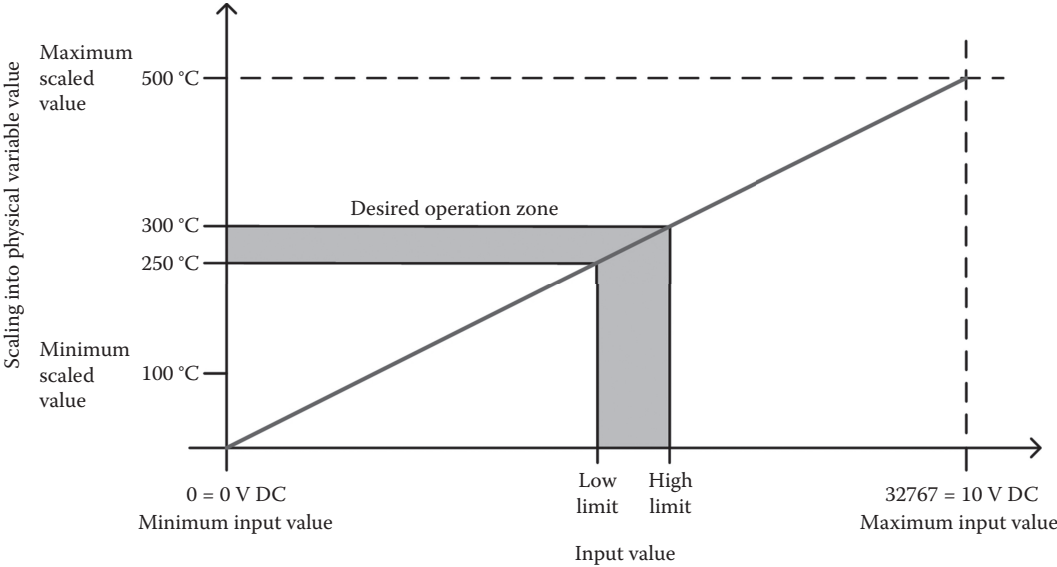


Figure B.2 Diagram for derivation of the scaling formula.

Applying Equation B.4 for the two temperature limit values, it is derived that,

$$\text{Low Limit} = (250 - 100) / (400 / 32767) = 12287$$

and

$$\text{High Limit} = (300 - 100) / (400 / 32767) = 16393$$

In some PLCs, system function blocks perform the required scaling operation. Otherwise, based on the above scaling equations, one has to program either the control of the temperature or any other simpler application as the indication of temperature in a digital panel, and then incorporate the required instructions to the whole PLC program of the automation.